What Is Claimed Is:

1   1.   A method for automatically propagating distributed components

2   during development of a distributed application, comprising:

3   receiving the distributed application;

4   automatically identifying under computer control any distributed

5   components within the distributed application that need to be deployed to remote

6   locations; and

7   for each distributed component that needs to be deployed to a remote

8   location, automatically deploying the distributed component under computer

9   control by,

10   identifying the remote location for the distributed

11   component, and

12   causing the distributed component to be deployed to the

13   remote location;

14   whereby a programmer of the distributed application does not have to

15   enter explicit commands to deploy distributed components to remote locations.

1   2.   The method of claim 1, further comprising:

2   automatically determining if any distributed components of the distributed

3   application have not been encapsulated as local components; and

4   for each distributed component that has not been encapsulated as a local

5   component, automatically encapsulating the distributed component as a local

6   component, so that the distributed component appears to be a local component.

1   3.   The method of claim 1, wherein identifying distributed

2   components that need to be deployed to remote locations involves examining a

18

3   deployment specifier that indicates where each of the distributed components that

4   make up the distributed application is to be deployed.

1   4.   The method of claim 1,

2   wherein the distributed application is specified in terms of a component-

3   behavior model;

4   wherein the component-behavior model specifies components, which are

5   separately deployable pieces of software that can be used to make up an

6   application; and

7   wherein the component-behavior model also specifies behaviors that

8   define a response to an event, wherein the response can include activating a

9   component.

1   5.   The method of claim 4, wherein activating the component involves

2   invoking a method defined by the component.

1   6.   The method of claim 4, wherein an event can be generated by a

2   component or a behavior.

1   7.   The method of claim 1,

2   wherein receiving the distributed application involves receiving the

3   distributed application after the distributed application has been modified during a

4   development process; and

5   wherein identifying distributed components that need to be deployed to

6   remote locations involves,

7   determining if any distributed components have been

8   modified, and then

19

1   determining where distributed components that have been

2   modified are to be deployed;

3   whereby only distributed components that have been modified during the

4   development process are deployed.

1   8.   The method of claim 1, wherein receiving the distributed

2   application involves:

3   authoring the distributed components that make up the distributed

4   application; and

5   creating a deployment specifier that indicates where each of the distributed

6   components is to be deployed.

1   9.   The method of claim 1, wherein receiving the distributed

2   application involves receiving the distributed application during execution of the

3   distributed application, wherein the distributed components that make up the

4   distributed application are not necessarily deployed prior to executing the

5   distributed application.

1   10.   The method of claim 1, wherein causing the distributed component

2   to be deployed to the remote location involves communicating with an application

3   server at the remote location through an administration protocol.

1   11.   The method of claim 1, wherein causing the distributed component

2   to be deployed to the remote location involves communicating with an application

3   server at the remote location through a deployment server at the remote location,

4   wherein the deployment server operates by:

5   halting an application server process;

20

6   loading files for the distributed component onto the application server;

7   setting preferences on the application server for the distributed component;

8   and

9   restarting the application server process.

1   12.   The method of claim 1, wherein each of the distributed

2   components can include one of:

3   an Enterprise JavaBean (EJB);

4   a Distributed Component Object Model (DCOM) object; and

5   a Common Object Request Broker Architecture (CORBA) object.

1   13.   The method of claim 1, wherein a distributed programming

2   component is an Enterprise JavaBean (EJB) that is encapsulated as a JavaBean by

3   combining functionality of a home interface and a remote interface of the EJB into

4   the JavaBean.

1   14.   The method of claim 1, further comprising:

2   determining a set of dependencies between distributed components that

3   make up the distributed application, wherein a dependency between a first

4   distributed component and a second distributed component indicates that the first

5   distributed component refers to the second distributed component; and

6   ensuring that each distributed component that depends on a remote

7   distributed component located on another computer system has a reference to the

8   remote distributed component.

1   15.   A method for deploying a component-behavior model within a

2   distributed computer system, comprising:

21

3   receiving a specification for the component-behavior model;

4   wherein the component-behavior model specifies components, which are

5   separately deployable pieces of software that can be used to make up an

6   application;

7   wherein the component-behavior model also specifies behaviors that

8   activate components in response to events generated by components or behaviors;

9   identifying components within the component-behavior model to be

10   deployed to remote locations; and

11   for each component to be deployed to a remote location,

12   identifying the remote location, and

13   causing the component to be deployed to the remote

14   location.

1   16.   The method of claim 15, wherein identifying components to be

2   deployed to remote locations involves examining a deployment specifier that

3   indicates where each of the components within the component-behavior model is

4   to be deployed.

1   17.   A computer-readable storage medium storing instructions that

2   when executed by a computer cause the computer to perform a method for

3   automatically propagating distributed components during development of a

4   distributed application, the method comprising:

5   receiving the distributed application;

6   automatically identifying under computer control any distributed

7   components within the distributed application that need to be deployed to remote

8   locations; and

22

9   for each distributed component that needs to be deployed to a remote

10   location, automatically deploying the distributed component under computer

11   control by,

12   identifying the remote location for the distributed

13   component, and

14   causing the distributed component to be deployed to the

15   remote location;

16   whereby a programmer of the distributed application does not have to

17   enter explicit commands to deploy distributed components to remote locations.

1   18.   The computer-readable storage medium of claim 17, wherein the

2   method further comprises:

3   automatically determining if any distributed components of the distributed

4   application have not been encapsulated as local components; and

5   for each distributed component that has not been encapsulated as a local

6   component, automatically encapsulating the distributed component as a local

7   component, so that the distributed component appears to be a local component.

1   19.   The computer-readable storage medium of claim 17, wherein

2   identifying distributed components that need to be deployed to remote locations

3   involves examining a deployment specifier that indicates where each of the

4   distributed components that make up the distributed application is to be deployed.

1   20.   The computer-readable storage medium of claim 17,

2   wherein the distributed application is specified in terms of a component-

3   behavior model;

4   wherein the component-behavior model specifies components, which are

5   separately deployable pieces of software that can be used to make up an

6   application; and

7   wherein the component-behavior model also specifies behaviors that

8   define a response to an event, wherein the response can include activating a

9   component.

1   21.   The computer-readable storage medium of claim 20, wherein

2   activating the component involves invoking a method defined by the component.

1   22.   The computer-readable storage medium of claim 20, wherein an

2   event can be generated by a component or a behavior.

1   23.   The computer-readable storage medium of claim 17,

2   wherein receiving the distributed application involves receiving the

3   distributed application after the distributed application has been modified during a

4   development process; and

5   wherein identifying distributed components that need to be deployed to

6   remote locations involves,

7   determining if any distributed components have been

8   modified, and then

9   determining where distributed components that have been

10   modified are to be deployed;

11   whereby only distributed components that have been modified during the

12   development process are deployed.

24

1   24.   The computer-readable storage medium of claim 17, wherein

2 receiving the distributed application involves:

3   authoring the distributed components that make up the distributed

4 application; and

5   creating a deployment specifier that indicates where each of the distributed

6 components is to be deployed.

1   25.   The computer-readable storage medium of claim 17, wherein

2 receiving the distributed application involves receiving the distributed application

3 during execution of the distributed application, wherein the distributed

4 components that make up the distributed application are not necessarily deployed

5 prior to executing the distributed application.

1   26.   The computer-readable storage medium of claim 17, wherein

2 causing the distributed component to be deployed to the remote location involves

3 communicating with an application server at the remote location through an

4 administration protocol.

1   27.   The computer-readable storage medium of claim 17, wherein

2 causing the distributed component to be deployed to the remote location involves

3 . communicating with an application server at the remote location through a

4 deployment server at the remote location, wherein the deployment server operates

5 by:

6   halting an application server process;

7   loading files for the distributed component onto the application server;

8   setting preferences on the application server for the distributed component;

9 and

10   restarting the application server process.

1   28.   The computer-readable storage medium of claim 17, wherein each

2   of the distributed components can include one of:

3   an Enterprise JavaBean (EJB);

4   a Distributed Component Object Model (DCOM) object; and

5   a Common Object Request Broker Architecture (CORBA) object.

1   29.   The computer-readable storage medium of claim 17, wherein a

2   distributed programming component is an Enterprise JavaBean (EJB) that is

3   encapsulated as a JavaBean by combining functionality of a home interface and a

4   remote interface of the EJB into the JavaBean.

1   30.   The computer-readable storage medium of claim 17, wherein the

2   method further comprises:

3   determining a set of dependencies between distributed components that

4   make up the distributed application, wherein a dependency between a first

5   distributed component and a second distributed component indicates that the first

6   distributed component refers to the second distributed component; and

7   ensuring that each distributed component that depends on a remote

8   distributed component located on another computer system has a reference to the

9   remote distributed component.

1   31.   A computer-readable storage medium storing instructions that

2   when executed by a computer cause the computer to perform a method for

3   deploying a component-behavior model within a distributed computer system, the

4   method comprising:

26

5   receiving a specification for the component-behavior model;

6   wherein the component-behavior model specifies components, which are

7   separately deployable pieces of software that can be used to make up an

8   application;

9   wherein the component-behavior model also specifies behaviors that

10 activate components in response to events generated by components or behaviors;

11   identifying components within the component–behavior model to be

12 deployed to remote locations; and

13   for each component to be deployed to a remote location,

14   identifying the remote location, and

15   causing the component to be deployed to the remote

16   location.

1   32.   The computer-readable storage medium of claim 31, wherein

2 identifying components to be deployed to remote locations involves examining a

3 deployment specifier that indicates where each of the components within the

4 component-behavior model is to be deployed.

1   33.   An apparatus that propagates distributed components during

2 development of a distributed application, comprising:

3   a receiving mechanism that is configured to receive the distributed

4 application

5   an identification mechanism that is configured to identify any distributed

6 components within the distributed application that need to be deployed to remote

7 locations; and

8   a deployment mechanism, wherein for each distributed component that

9   needs to be deployed to a remote location, the deployment mechanism is

10   configured to,

11   identify the remote location for the distributed component,

12   and to

13   cause the distributed component to be deployed to the

14   remote location;

15   whereby a programmer of the distributed application does not have to

16   enter explicit commands to deploy distributed components to remote locations.

1   34.   The apparatus of claim 33, further comprising an encapsulation

2   mechanism that is configured to determine if any distributed components of the

3   distributed application have not been encapsulated as local components;

4   wherein for each distributed component that has not been encapsulated as

5   a local component, the encapsulation mechanism is configured to encapsulate the

6   distributed component as a local component, so that the distributed component

7   appears to be a local component.

1   35.   The apparatus of claim 33, wherein the identification mechanism is

2   additionally configured to examine a deployment specifier that indicates where

3   each of the distributed components that make up the distributed application is to

4   be deployed.

1   36.   The apparatus of claim 33,

2   wherein the distributed application is specified in terms of a component-

3   behavior model;

28

4   wherein the component-behavior model specifies components, which are

5   separately deployable pieces of software that can be used to make up an

6   application; and

7   wherein the component-behavior model also specifies behaviors that

8   define a response to an event, wherein the response can include activating a

9   component.

1   37.   The apparatus of claim 36, wherein activating the component

2   involves invoking a method defined by the component.

1   38.   The apparatus of claim 36, wherein an event can be generated by a

2   component or a behavior.

1   39.   The apparatus of claim 33,

2   wherein the receiving mechanism is configured to receive the distributed

3   application after the distributed application has been modified during a

4   development process; and

5   wherein the identification mechanism is configured to,

6   determine if any distributed components have been

7   modified, and then to

8   determine where distributed components that have been

9   modified are to be deployed;

10   whereby only distributed components that have been modified during the

11   development process are deployed.

1   40.   The apparatus of claim 33, further comprising an authoring

2   mechanism that is configured to facilitate authoring the distributed components

29

3   that make up the distributed application, and to facilitate creating a deployment

4   specifier that indicates where each of the distributed components is to be

5   deployed.

1   41.   The apparatus of claim 33, wherein the receiving mechanism is

2   configured to receive the distributed application during execution of the

3   distributed application, wherein the distributed components that make up the

4   distributed application are not necessarily deployed prior to executing the

5   distributed application.

1   42.   The apparatus of claim 33, wherein the deployment mechanism is

2   configured to communicate with an application server at the remote location

3   through an administration protocol.

1   43.   The apparatus of claim 33, wherein the deployment mechanism is

2   configured to communicate with an application server at the remote location

3   through a deployment server at the remote location, wherein the deployment

4   server is configured to:

5   halt an application server process;

6   load files for the distributed component onto the application server;

7   set preferences on the application server for the distributed component;

8   and to

9   restart the application server process.

1   44.   The apparatus of claim 33, wherein each of the distributed

2   components can include one of:

3   an Enterprise JavaBean (EJB);

30

4   a Distributed Component Object Model (DCOM) object; and

5   a Common Object Request Broker Architecture (CORBA) object.

1   45.   The apparatus of claim 33, wherein each distributed programming

2   component is an Enterprise JavaBean (EJB) that is encapsulated as a JavaBean by

3   combining functionality of a home interface and a remote interface of the EJB into

4   the JavaBean.

1   46.   The apparatus of claim 33, further an interlinking mechanism that

2   is configured to:

3   determine a set of dependencies between distributed components that

4   make up the distributed application, wherein a dependency between a first

5   distributed component and a second distributed component indicates that the first

6   distributed component refers to the second distributed component; and to

7   ensure that each distributed component that depends on a remote

8   distributed component located on another computer system has a reference to the

9   remote distributed component.

1   47.   An apparatus that deploys a component-behavior model within a

2   distributed computer system, comprising:

3   a receiving mechanism that is configured to receive a specification for the

4   component-behavior model;

5   wherein the component-behavior model specifies components, which are

6   separately deployable pieces of software that can be used to make up an

7   application;

8   wherein the component-behavior model also specifies behaviors that

9   activate components in response to events generated by components or behaviors;

31

10   an identification mechanism that is configured to identify components

11   within the component-behavior model to be deployed to remote locations; and

12   a deployment mechanism, wherein for each component to be deployed to a

13   remote location, the deployment mechanism is configured to,

14   identify the remote location, and to

15   cause the component to be deployed to the remote location.

1   48.   The apparatus of claim 47, wherein the identification mechanism is

2   configured to examine a deployment specifier that indicates where each of the

3   components within the component-behavior model is to be deployed.

32